

공개SW 솔루션 설치 & 활용 가이드

미들웨어 > WEB 서버



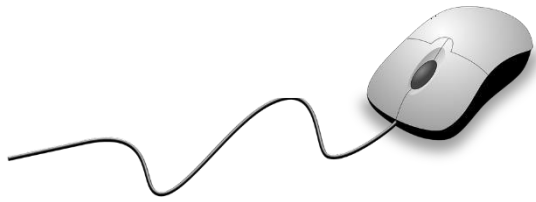
제대로 배워보자

How to Use Open Source Software

Open Source Software Installation & Application Guide



오픈소스 소프트웨어 통합지원센터
Open Source Software Support Center



CONTENTS

1. 개요
2. 기능요약
3. 실행환경
4. 설치 및 실행
5. 기능소개
6. 활용예제
7. FAQ
8. 용어정리

1. 개요



<p>소개</p>	<ul style="list-style-type: none"> • 1995년 웹 서버중 하나인 NCSA httpd 1.3 웹서버를 기반으로 탄생 (National Center for Supercomputing Applications: 미국국립수퍼컴퓨터활용센터) • 아파치 소프트웨어 재단 (Apache Software Foundation) 에서 개발하여 배포하는 공개SW 웹서버 소프트웨어 		
<p>주요기능</p>	<ul style="list-style-type: none"> • 웹 서버의 정적 콘텐츠를 효율적으로 제공 (CSS, HTML) • 다양한 보안 기능 (SSL, Proxy, ACL, Directory 접근제한) • 하나의 서버에서 여러 도메인 운영 (가상호스트 기능) 		
<p>대분류</p>	<ul style="list-style-type: none"> • 미들웨어 	<p>소분류</p>	<ul style="list-style-type: none"> • Web Server
<p>라이선스형태</p>	<ul style="list-style-type: none"> • Apache License v2.0 	<p>사전설치 솔루션</p>	<ul style="list-style-type: none"> • 없음
		<p>버전</p>	<ul style="list-style-type: none"> • 2.4.28 (2017년 10월 기준)
<p>특징</p>	<ul style="list-style-type: none"> • 안정성과 성능으로 확장성으로 세계에서 가장 많이 사용되는 웹 서버 • 공개SW로 무료로 사용가능하며 다른 상용 웹서버와 비슷한 성능 보유 • 지속적인 패치파일 제공으로 최신의 업그레이드가 가능 • 다양한 추가 기능이 있으며 구축 용이, 유연한 모듈 지원 		
<p>보안취약점</p>	<ul style="list-style-type: none"> • 취약점 ID : CVE-2017-9798 • 심각도 : 7.5 HIGH(V3) • 취약점 설명 : '.htaccess'파일에 Limit 지시문이 설정되어 있고 유효하지 않은 HTTP 메서드로 설정된 시스템에서 원격 사용자는 경로에 대해 특수하게 조작 된 HTTP OPTIONS 요청을 보내 메모리 오류를 트리거하고 잠재적으로 민감한 정보 획득 • 대응방안 : 2.4.27 이상으로 업그레이드 • 참고 경로 : https://security.gentoo.org/glsa/201710-32 		
<p>개발회사/커뮤니티</p>	<ul style="list-style-type: none"> • Apache Software Foundation 		
<p>공식 홈페이지</p>	<ul style="list-style-type: none"> • https://www.apache.org 		



2. 기능요약



- Apache 주요 기능

보안기능	<ul style="list-style-type: none">• SSL, Proxy, ACL, Directory 접근제한* Directory 접근제한 : 웹서버가 제공하는 페이지, 경로에 대한 접근제한 설정
성능적인 측면	<ul style="list-style-type: none">• 리소스 분산처리, Cache(Expires), HTTP 표준설정• MPM(Multi-Processing Module), KeepAlive
가상호스트 기능	<ul style="list-style-type: none">• 하나의 서버에 여러 도메인 운영, 서버호스팅
운영적인 측면	<ul style="list-style-type: none">• ErrorDocument, 지시자를 사용해 특정 에러 발생 시 특정 페이지로 Redirect• 사용자에게 편리하고 친절한 메시지 표시• Access Log 가 기록되어 접속한 클라이언트의 IP, 도메인 등 접속정보 분석 가능
부가적인 기능	<ul style="list-style-type: none">• 유용한 Apache Module 등 제공

3. 실행환경



- 하드웨어 제약이 거의 없음
- OS 플랫폼 종류에 따른 지원
 - Unix 계열
 - 비Unix 플랫폼 : BeOS, OS/2, Windows
- 요구사항
 - 디스크공간: 최소 50MB 이상(설치시 아파치는 약 10MB 디스크 공간 차지)
 - ANSI-C 컴파일러와 컴파일 시스템 설치 확인
 - 정확한 시간 표시(Network Time Protocol 기반 ntpdate or xntpd 사용)



4. 설치 및 실행

세부 목차

1. 설치 및 실행(Windows)
2. 설치 및 실행(CentOS – RPM)
3. 설치 및 실행(CentOS – Source)



4. 설치 및 실행



4.1 설치 및 실행(Windows)(1/4)

- 설치 파일 버전
 - OS Version : Windows Server 2012 R2
 - Visual Studio 2017 (VC_redist.x64)
 - Apache Version : httpd-2.4.28-Win64-VC15.zip

- 2017년 10월 기준 공식 사이트에서 2.2.34 소스버전 파일만 제공하기 때문에 아래 사이트에 미리 컴파일 되어있는 파일 사용

* <https://www.apachelounge.com/download/>

1. 아파치 설치 전 해당 사이트에 있는 Visual Studio 2017 파일 다운/설치

2. OS Version, Bit 에 맞는 Apache 파일 다운, 압축 해제



4. 설치 및 실행



4.1 설치 및 실행(Windows)(2/4)

3. 압축해제 후 나오는 Apache24 폴더를 원하는 곳에 이동 (권장 : C:\W)

4. Apache설치경로\conf\httpd.conf 수정

1. ServerRoot 경로를 설치한 경로로 변경

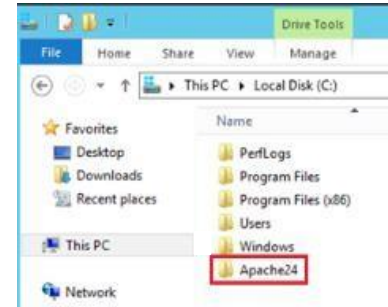
ServerRoot " c:/Apache24" ← ServerRoot "설치경로"

* 여기서는 기본경로 사용하며, 디렉터리 기호인 \는 가급적 / 로 변경 (php 설치할 경우 마찬가지로 적용)

4.2 포트 변경

Listen 80 ← Listen "사용할 포트"

* 여기서는 기본포트 사용



[C 드라이브 압축해제]

```
# Do not add a slash at the end of the directory
# ServerRoot at a non-local disk, be sure to use the
# Mutex directive, if file-based mutexes are used
# same ServerRoot for multiple httpd daemons,
# use the Mutex directive.
#
ServerRoot "c:/Apache24"

#
# Mutex: Allows you to set the mutex mechanism
# for individual mutexes, or change the global
# mutex file directory.
# Uncomment and change the directory if mutex
# mutex file directory is not on a local disk
# or for other reason.
#
# Mutex default:logs

#
# Listen: Allows you to bind Apache to specific
# ports, instead of the default. See also the
# Listen directive.
#
# Change this to Listen on specific IP addresses
# to prevent Apache from glomming onto all bound
# ports.
#
#Listen 12.34.56.78:80
Listen 80
```

[httpd.conf수정]



4. 설치 및 실행



4.1 설치 및 실행(Windows)(3/4)

4.3 문서 저장위치 변경

- DocumentRoot "c:/Apache24/htdocs" ←
DocumentRoot "웹문서루트"
- <Directory "c:/Apache24/htdocs"> ←
<Directory " 웹문서경로 " >

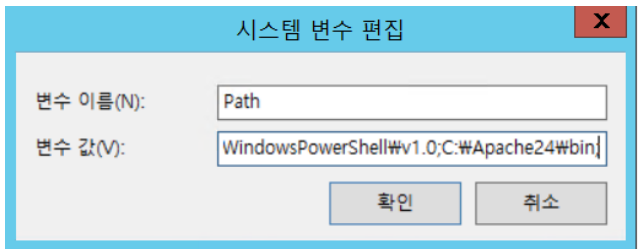
```
#
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
DocumentRoot "c:/Apache24/htdocs"
<Directory "c:/Apache24/htdocs">
#
# Possible values for the Options directive are "None", "All",
# or any combination of:
#   Indexes Includes FollowSymLinks SymLinksifOwnerMatch ExecCGI MultiViews
#
# Note that "MultiViews" must be named *explicitly* --- "Options All"
# doesn't give it to you.
#
# The Options directive is both complicated and important. Please see
# http://httpd.apache.org/docs/2.4/mod/core.html#options
# for more information.
#
Options Indexes FollowSymLinks

#
# AllowOverride controls what directives may be placed in .htaccess files.
# It can be "All", "None", or any combination of the keywords:
#   AllowOverride FileInfo AuthConfig Limit
#
AllowOverride None

#
# Controls who can get stuff from this server.
#
Require all granted
</Directory>
```

[httpd.conf 수정]

5. + R 누른 후 control 입력
6. 시스템 > 고급 시스템 설정 > 고급 탭의 "환경 변수(N)"
단추 클릭 > 시스템 변수 > Path 변수 편집 > 맨 끝에
설치경로 추가 (설치경로는 " ; " 세미콜론 으로 구분)




[환경변수 수정]



4. 설치 및 실행



4.1 설치 및 실행(Windows)(4/4)

7.  + X 누른 후 명령 프롬프트를 관리자 권한으로 실행
8. "httpd -k install" 입력
9. "httpd -k start" 입력
10. 웹브라우저에서 <http://localhost> 입력 후 It works! 확인

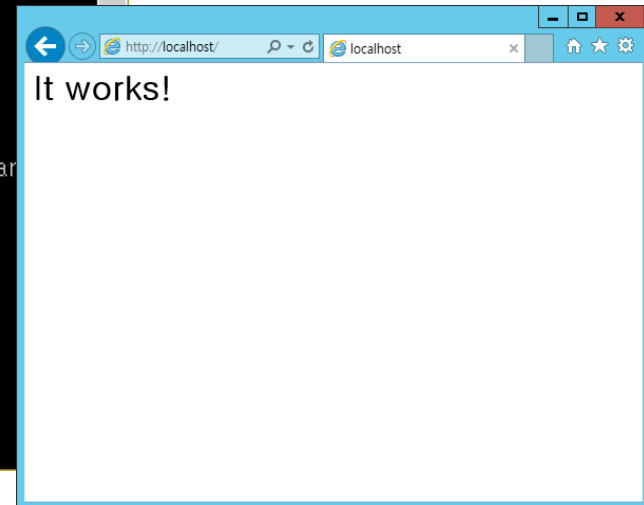
```
관리자: 명령 프롬프트
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Windows\system32>httpd -k install
Installing the 'Apache2.4' service
The 'Apache2.4' service is successfully installed.
Testing httpd.conf...
Errors reported here must be corrected before the service can be started.

C:\Windows\system32>httpd -k start

C:\Windows\system32>
```

[아파치 시작]



[웹브라우저 확인]



4. 설치 및 실행



4.2 설치 및 실행(Linux - RPM)(1/3)

- OS Version : CentOS Linux release 7.4.1708 Apache Version : httpd-2.4.6-67.el7.centos.5.x86_64
- CentOS 에서는 기본적으로 yum 을 사용하여 RPM 설치 가능

• Yum 이란?

RPM 기반의 시스템을 위한 자동 업데이트 및 패키지 설치/제거 도구 (의존성 패키지를 같이 설치)

1. 서버에 httpd 패키지가 설치되어 있는지 확인

```
# rpm -qa | grep httpd
```

```
[root@localhost ~]# yum info httpd
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: www.ftp.ne.jp
 * extras: www.ftp.ne.jp
 * updates: centos.mirror.cdnetworks.com
Available Packages
Name      : httpd
Arch      : x86_64
Version   : 2.4.6
Release   : 67.el7.centos.5
Size      : 2.7 M
Repo      : updates/7/x86_64
Summary   : Apache HTTP Server
URL       : http://httpd.apache.org/
License   : ASL 2.0
Description : The Apache HTTP Server is a powerful, efficient, and
           : web server.
```

[설치 가능한 httpd 패키지 버전 확인]

2. 설치 가능한 httpd 패키지 버전 확인

```
# yum info httpd
```

3. 패키지 설치 및 확인

```
# yum install httpd -y
```

```
# rpm -qa | grep httpd
```

```
[root@localhost ~]# rpm -qa | grep httpd
httpd-tools-2.4.6-67.el7.centos.5.x86_64
httpd-2.4.6-67.el7.centos.5.x86_64
```

[패키지 설치 후 확인]



4. 설치 및 실행



4.2 설치 및 실행(Linux - RPM)(2/3)

4. 서비스 시작

```
# systemctl start httpd
```

5. 서비스 상태 확인

```
# systemctl status httpd
```

6. 재부팅 시 자동시작

```
# systemctl enable httpd
```

```
# systemctl status httpd
```

* 위 httpd status Loaded 행에서 enabled 확인

* vendor preset : 패키지 초기값

```
[root@localhost ~]# systemctl start httpd
[root@localhost ~]#
[root@localhost ~]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
   Active: active (running) since Mon 2017-10-16 20:29:54 KST; 1min 47s ago
     Docs: man:httpd(8)
           man:apachectl(8)
  Main PID: 2899 (httpd)
    Status: "Processing requests..."
    CGroup: /system.slice/httpd.service
            └─2899 /usr/sbin/httpd -DFOREGROUND
              └─2902 /usr/sbin/httpd -DFOREGROUND
                └─2903 /usr/sbin/httpd -DFOREGROUND
                  └─2904 /usr/sbin/httpd -DFOREGROUND
                    └─2905 /usr/sbin/httpd -DFOREGROUND
                      └─2906 /usr/sbin/httpd -DFOREGROUND

Oct 16 20:29:54 localhost.localdomain systemd[1]: Starting The Apache HTTP Server: httpd.
Oct 16 20:29:54 localhost.localdomain httpd[2899]: AH00558: httpd: could not open error log file: [error] [warn]
Oct 16 20:29:54 localhost.localdomain systemd[1]: Started The Apache HTTP Server: httpd.
Hint: Some lines were ellipsized, use -l to show in full.
```

[서비스 시작 및 확인]

```
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2017-10-16 20:29:54 KST; 8min ago
```

[서비스 자동시작 설정 확인]



4. 설치 및 실행

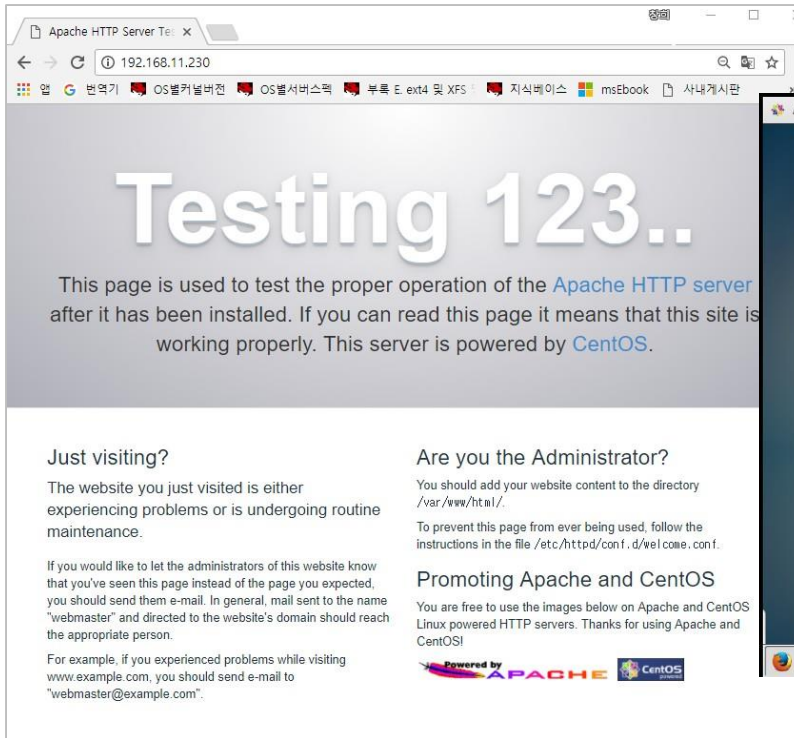


4.2 설치 및 실행(Linux - RPM)(3/3)

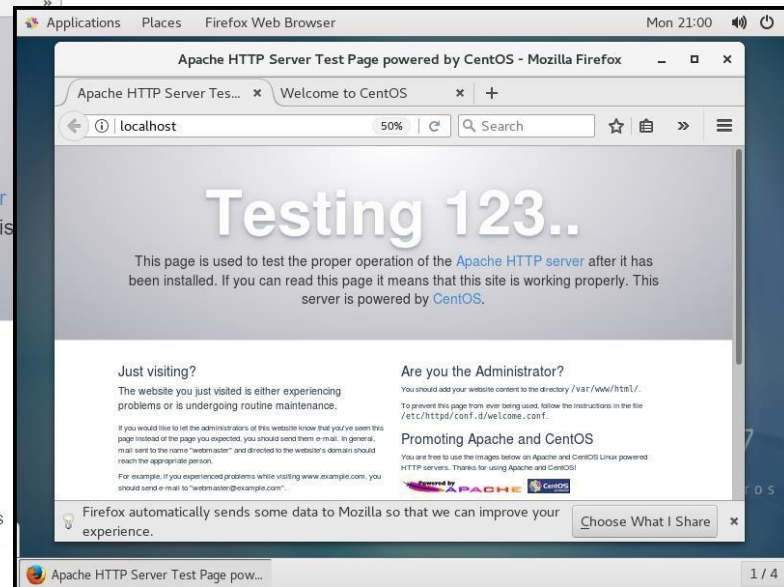
7. 사이트 확인

웹브라우저에서 <http://서버IP> 로 테스트 페이지 접속 확인

- * 리눅스 서버가 GUI (Graphic User Interface) 모드 일 경우 서버에서 웹브라우저로 확인 가능



[클라이언트에서 사이트 확인]



[리눅스 GUI 모드에서 사이트 확인]

4. 설치 및 실행

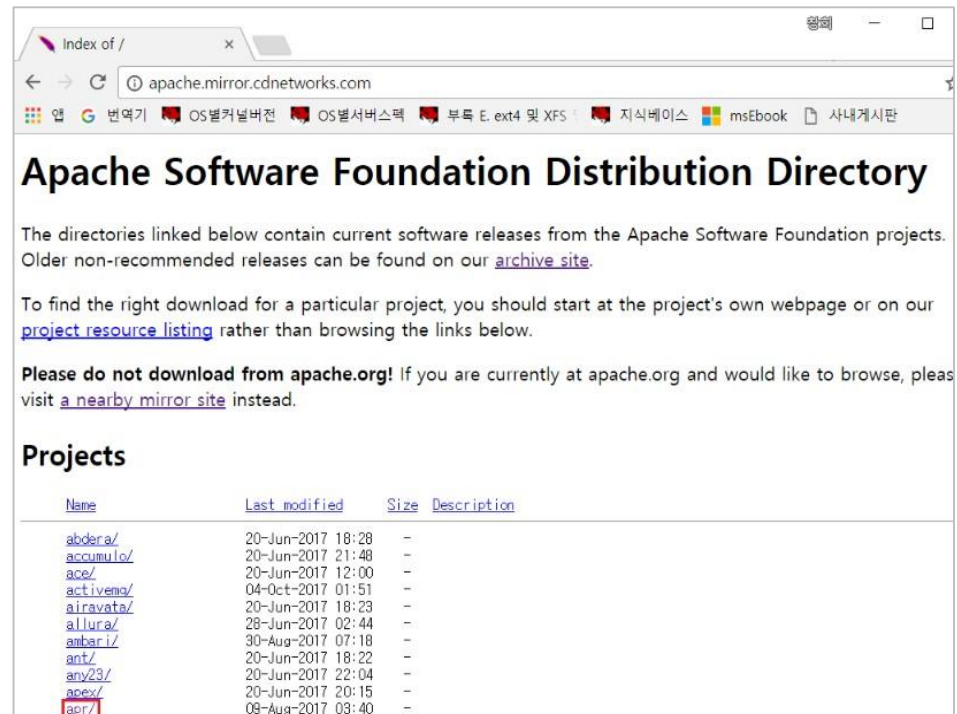


4.3 설치 및 실행(CentOS – Source)(1/5)

- 2017년 10월 기준 OS, 아파치 최신버전으로 설치
 - OS : CentOS 7.4 64bit
 - 아파치 파일 : httpd-2.4.28.tar.gz
 - 필요패키지 : apr-1.6.2.tar.gz, apr-util-1.6.0.tar.gz, pcre-8.41.tar.gz

•다운로드 경로 <http://apache.mirror.cdnetworks.com/httpd/> <http://apache.mirror.cdnetworks.com/apr/> <https://ftp.pcre.org/pub/pcre/>

1. 서버 /usr/local/src 에 파일 다운로드



[아파치 공식 미러사이트]



4. 설치 및 실행



4.3 설치 및 실행(CentOS – Source)(2/5)

2. 사전설치

Requirement

- * 해당 패키지가 없을 경우 컴파일 중 에러발생
yum -y install gcc openssl-devel expat-devel
사전설치 패키지는 의존성으로 설치해야 하는 패키지가 많고 복잡하므로 여기서는 Yum 으로 설치)

3. 다운로드한 파일 압축 해제

- # cd /usr/local/src
- # tar xvzf apr-1.6.2.tar.gz
- # tar xvzf apr-util-1.6.0.tar.gz
- # tar xvzf httpd-2.4.28.tar.gz
- # tar xvzf pcre-8.41.tar.gz
- # mv apr-1.6.2 ./httpd-2.4.28/src/lib/apr
- # mv apr-util-1.6.0 ./httpd-2.4.28/src/lib/apr-util

```
Dependencies Resolved
-----
Package                               Arch                               Version
-----
Installing:
  expat-devel                          x86_64                             2.1.0-10.el7
  openssl-devel                         x86_64                             1:1.0.2k-8.el7
Installing for dependencies:
  keyutils-libs-devel                  x86_64                             1.5.8-3.el7
  krb5-devel                            x86_64                             1.15.1-8.el7
  libcom_err-devel                     x86_64                             1.42.9-10.el7
  libkadm5                              x86_64                             1.15.1-8.el7
  libsasl-devel                         x86_64                             2.5-11.el7
  libsepol-devel                       x86_64                             2.5-6.el7
  libverto-devel                       x86_64                             0.2.5-4.el7
  pcre-devel                            x86_64                             8.32-17.el7
  zlib-devel                            x86_64                             1.2.7-17.el7
-----
Transaction Summary
-----
Install 2 Packages (+9 Dependent packages)
```

[의존성 패키지 설치 (yum)]

```
[root@localhost src]# rpm -qa | egrep "gcc|openssl-devel|expat-devel"
gcc-4.8.5-16.el7.x86_64
libgcc-4.8.5-16.el7.x86_64
openssl-devel-1.0.2k-8.el7.x86_64
gcc-c++-4.8.5-16.el7.x86_64
gcc-gfortran-4.8.5-16.el7.x86_64
expat-devel-2.1.0-10.el7_3.x86_64
```

[패키지 설치 확인]



4. 설치 및 실행



4.3 설치 및 실행(CentOS – Source)(3/5)

4. pcre 설치

```
# cd /usr/local/src/pcre-8.41
# ./configure
# make
# make install
```

(컴파일 결과에 에러유무 확인)

```
config.status: creating build/config_vars.sh
config.status: creating include/ap_config_auto.h
config.status: executing default commands
configure: summary of build options:

Server Version: 2.4.28
Install prefix: /usr/local/apache2
C compiler:      gcc -std=gnu99
CFLAGS:         -g -O2 -pthread
LDFLAGS:
LIBS:
CPPFLAGS:       -DLINUX -D_REENTRANT -D_GNU_SOURCE
C preprocessor: gcc -E
```

[httpd configure 결과]

5. httpd 설치

```
# cd /usr/local/src/httpd-2.4.28/
# ./configure --prefix=/usr/local/httpd-2.4.28 --
  enable-mods-shared=most --enable-rewrite --
  enable-ssl --enable-expirer
# make
# make install
```

* PCRE 란?

Perl 호환 정규표현식

Perl 5와 같은 구문, 의미를 사용하여 정규식 패턴일치 구현하는 함수 집합
메일 전송 에이전트 용으로 만들었으나 자체 API, POSIX 호환성 같은
강점으로 오픈소스 프로젝트에서 두루두루 사용



4. 설치 및 실행



4.3 설치 및 실행(CentOS – Source)(4/5)

6. 아파치 시작 및 확인

```
# /usr/local/httpd-2.4.28/bin/httpd -k start
```

```
# ps -ef | grep httpd | grep -v grep
```

 ← 아파치 실행프로세스 확인

```
# netstat -anp | grep httpd
```

 ← 아파치 서비스 포트 확인

```
# curl http://127.0.0.1
```

 ← 127.0.0.1 은 서버 자신을 가르키는 주소로
서버 http 아파치 연결 시 페이지 출력 확인

```
[root@localhost ~]# /usr/local/httpd-2.4.28/bin/httpd -k start
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using loc
[root@localhost ~]# ps -ef | grep httpd | grep -v grep
root      40305      1  0 20:46 ?        00:00:00 /usr/local/httpd-2.4.28/bin/httpd -k start
daemon    40306  40305  0 20:46 ?        00:00:00 /usr/local/httpd-2.4.28/bin/httpd -k start
daemon    40307  40305  0 20:46 ?        00:00:00 /usr/local/httpd-2.4.28/bin/httpd -k start
daemon    40308  40305  1 20:46 ?        00:00:00 /usr/local/httpd-2.4.28/bin/httpd -k start
[root@localhost ~]# netstat -anp | grep httpd
tcp6      0      0 :::80          :::*           LISTEN      40305/httpd
[root@localhost ~]# curl http://127.0.0.1
<html><body><h1>It works!</h1></body></html>
```

[아파치 시작 및 확인]



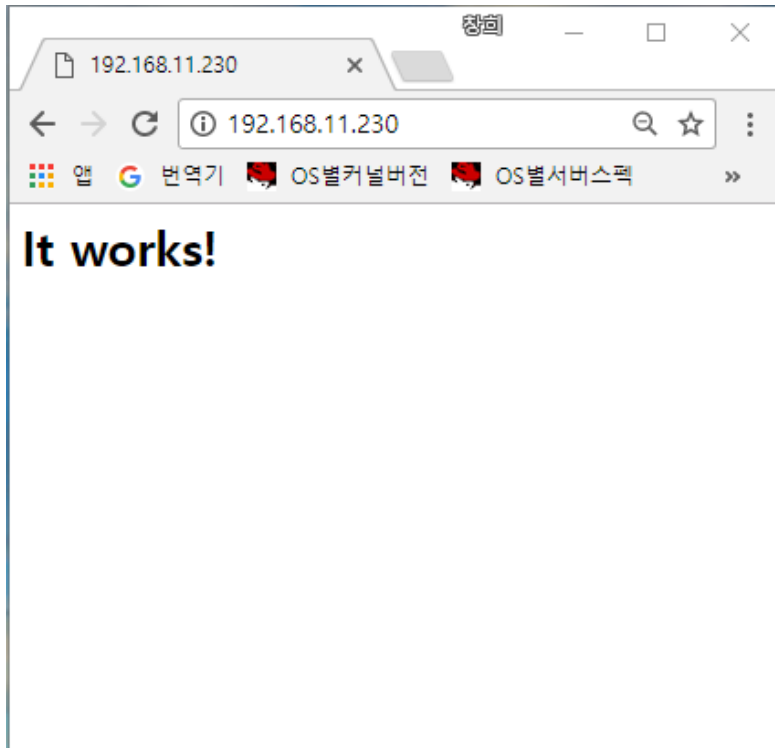
4. 설치 및 실행



4.3 설치 및 실행(CentOS – Source)(5/5)

참고 : httpd start 시 "httpd: could not reliably determine the server's fully qualified domain name~" 문구가 출력되는데, 설정파일에서 ServerName 옵션값을 설정하지 않아서 나오는 구문으로 정확한 도메인명을 입력 요청하는 구문이다. 오류 구문이 아니며 이 문서에서는 해당 구문을 무시한다.

7. 웹브라우저에서 사이트 확인



[웹브라우저 확인]

5. 기능소개

세부 목차

1. httpd.conf 설정 파일
2. httpd 다양한 옵션 및 기능
3. 가상호스트 기능
4. 가상호스트 설정



5. 기능소개



5.1 httpd.conf 설정 파일(1/8)

httpd.conf 파일은 아파치의 주설정파일이다.

다양한 옵션과 설정으로 뛰어난 보안과 성능 향상이 장점

- `ServerRoot "/usr/local/apache"`

아파치의 Root 디렉터리. 이후 대부분의 경로(PATH)들은 이 경로를 Root로 한 상대경로로 표시됨

- `Listen 80`

아파치에서 사용하는 default 포트

80 포트 이외 다른 포트 사용 가능

- `ServerAdmin root@localhost`

여기서 지정하는 email 주소는 웹문서 로딩 시 에러가 발생할 경우 에러페이지에 보여질 아파치 웹서버 관리자의 email 주소

- `ServerName www.example.com`

클라이언트에게 보여주는 호스트이름, 예로 www를 쓰지 않는 호스트에서 www를 쓰는 것처럼 보이게 할 수 있다. 현재 사용하는 도메인이 없다면 IP 로 사용할 수 있다.



5. 기능소개



5.1 httpd.conf 설정 파일(2/8)

- `DocumentRoot "/var/www/html"`

웹문서의 루트가 있는 경로를 지정함

가상호스트 설정에서 이 지시자를 이용하여 다른 경로에 있는 웹문서를 지정할 수 있다.

- `<Directory />`
 `Options FollowSymLinks`
 `AllowOverride None Or`
 `der deny,allow`
 `Deny from all`
`</Directory>`

`<Directory>` 지시자는 디렉터리를 제어하기 위한 구문.

`<Directory>` ~ `</Directory>` 안에는 디렉터리를 제어하는 모든 지시어를 사용할 수 있음

Options : 접근제어를 설정, 해당 옵션에는 아래값이 있다.

- None : 모두 차단
- All : 모두 허용
- Indexes : DirectoryIndex 옵션에서 설정한 파일 index.html, index.php 파일이 존재하지 않을 경우 웹 브라우저로 디렉터리내 파일목록 리스트를 표시 → 파일리스트 노출시킴으로 보안상 좋지 않다.



5. 기능소개



1. httpd.conf 설정 파일(3/8)

- FollowSymlinks : 심볼릭 링크 허용, 웹브라우저에서 실제 심볼릭 링크파일 경로까지 확인 가능하므로 보안상 좋지 않다.
- ExecCGI : perl 등과 같은 CGI 실행을 허용, perl 외에도 CGI 실행이 되지 않을 경우 해당 옵션을 확인해야 한다.

AllowOverride : 접근방법 결정, 설정하는 값들은 중복해서 설정될 수 있으며 가장 최근 설정값이 적용됨

- None : AccessFileName 에 지정된 파일을 액세스 인증파일로 인식하지 않음, 보통 AccessFileName 값은 .htaccess 이므로 이를 무시하여 새로운 접근인증방식을 Override 하지 않음
- All : 이전의 인증방식에 대하여 새로운 접근인증방식을 우선 적용하도록 Override를 허용한다.
- AuthConfig : AccessFileName 지시자에 명시한 파일 .htaccess 에 대하여 다양한 클라이언트 인증지시자의 사용을 허용, htpasswd 명령어로 특정 디렉터리 접근을 .htaccess 파일로 제어
- FileInfo : AccessFileName 지시자에 명시한 파일에 대하여 문서유형을 제어하는 지시자 허용 (ErrorDocument)
- Indexes : AccessFileName 지시자에 명시한 파일에 대해 디렉터리 Indexing 제어 지시자 허용
(AddDescription, DirectoryIndex, FancyIndexing, IndexOptions)
- Limit : AccessFileName 지시자에 명시한 파일에 대해 호스트접근 제어 지시자 허용 (allow, deny, order)
- Options : AccessFileName 지시자에 명시한 파일에 대해 Options, XBitHack 같은 특정 디렉터리 옵션을 제어하는 지시자 허용



5. 기능소개



5.1 httpd.conf 설정 파일(4/8)

- ErrorLog "logs/error_log"

웹서버 에러로그파일 지정 가상호스트 설정에서 별도 에러로그파일을 사용하지 않을 경우 해당 로그파일 사용

- LogLevel warn

ErrorLog 파일 기록 내용을 얼마나 자세하게 또는 어떤 수준까지 기록할 것인가를 결정

해당 지시자 LogLevel은 아래와 같다. (아래로 갈수록 세부적으로 출력)

- none : 발생하는 모든 상황에 대한 메시지 무시
- emerg : emergency, 최상위 매우 위험한 상황에 대한 메시지, 전체 공지가 요구됨
- alert : 즉각적인 조치를 취해야하는 상황의 메시지
- crit : 급한 상황은 아니지만 critical한(치명적인) 메시지로 빠른 조치 필요
- error : 에러가 발생한 상황 메시지
- warn : 주의를 요구하는 메시지, 경고상황 (warning)
- notice : 알림, 보통 수준보다 중요한 상황
- info : 단순한 프로그램에 대한 정보(information) 및 메시지
- debug : 최하위, 디버깅 관련 메시지
- * : 발생하는 모든 상황에 대한 메시지

→ 오류 로그를 너무 자세하게 표시할 경우 불필요한 내용이 방대해지며 적절한 수준 설정 필요

5. 기능소개



5.1 httpd.conf 설정 파일(5/8)

- `Redirect permanent /foo http://www.example.com/bar`

특정 디렉터리 경로로 접근 시 다른 웹사이트경로(URL)로 Redirect

위의 예로 보면 /foo 디렉터리 접근 시 <http://www.example.com/bar> Redirect

- `ScriptAlias /cgi-bin/ "/usr/local/apache/cgi-bin/"`

CGI 스크립트를 특정 경로 /usr/local/apache/cgi-bin 에서만 제공

보안향상 옵션으로 사용자가 해당 경로에 존재하는 cgi 파일만 사용할 수 있도록 한다.

- `DefaultType text/plain`

Mime.types 파일에 정의되지 않은 파일형식에 대한 요청을 받았을 때 알 수 없는 문서타입에 대해 사용할 기본적인 mime 타입, 위의 내용도 지정되어 있다면 아파치에서 인식하지 못하는 mime 타입에 대한 요청을 받았을 때 일반 텍스트 문서로 인식하여 로딩한다.

- `TypesConfig conf/mime.types`

Mime.types 파일을 어디서 찾을 것인가를 지정

Mimye.types 파일은 서버에 의해 리턴될 수 있는 파일명과 mime 형식을 기술



5. 기능소개



1. httpd.conf 설정 파일(6/8)

- # prefork MPM

전통적인 아파치 요청처리 방식

<IfModule mpm_prefork_module>

StartServers	Min	5	: 아파치 데몬 시작 시 자식프로세스 개수
nSpareServers		5	: 아파치의 빠른 응답속도를 위한 유틸서버 최소개수, 해당 값보다 적으면 추가생성
MaxSpareServers		10	: 유틸서버 최대개수, 해당 값보다 많으면 유틸서버 삭제
MaxClients		150	: 초기 시작 시 실행가능한 최대 아파치 자식프로세스 개수
MaxRequestsPerChild		0	: 자식프로세스들이 처리할 수 있는 클라이언트 요청개수, 해당 값만큼 요청을 받으면 자식프로세스는 죽음 0 설정 시 죽지 않음 ← 부하, 자원낭비 면에서 적절한 사용 필요

- # worker MPM

방문자의 요청을 스레드 방식으로 처리

<IfModule mpm_worker_module>

StartServers		2	: 아파치 데몬 시작 시 서퍼 프로세스 개수
MaxClients		150	: 동시 연결가능한 클라이언트의 최대개수



5. 기능소개



5.1 httpd.conf 설정 파일(7/8)

prefork 모듈의 MaxClients 옵션과는 그 의미가 다름,
지정한 값(연결개수) 이상의 클라이언트 요청이 생긴다면
아파치는 응답하지 않고 클라이언트 요청을 무시프로세스가
시스템 자원을 무한정 차지하는 것 방지

MinSpareThreads	25	: 새로운 요청을 지체없이 처리하기 위해 대기하고 있는 최소 스레드 개수
MaxSpareThreads	75	: 새로운 요청을 지체없이 처리하기 위해 대기하고 있는 최대 스레드 개수
ThreadsPerChild	25	: 자식프로세스가 가질 수 있는 스레드 개수
MaxRequestsPerChild	0	: 자식프로세스가 서비스할 수 있는 최대 요청개수

</IfModule>

- UserDir public_html

웹서버에서 여러 사용자의 홈페이지를 각각 만들어 사용 시 필요한 개별 사용자의 홈페이지 디렉터리
이름 예로 test 라는 계정의 홈페이지가 <http://www.test.co.kr/~test> 일 때 test 계정의 "public_html" 디
렉터리가 홈디렉터리가 되어 이 디렉터리에 있는 초기문서 index.html 을 출력



5. 기능소개



1. httpd.conf 설정 파일(8/8)

<http://www.test.co.kr/~test> 접속 시 연결 및 페이지출력 상세과정

- 1) DNS 로 www.test.co.kr 해석 후 서버에 요청
- 2) 서버 /etc/passwd 파일에서 test 계정 홈디렉터리 /home/test 확인
- 3) UserDir 지시자에 지정된 디렉터리 public_html 확인
- 4) /home/test/public_html 디렉터리에서 DirectoryIndex 에 나열된 순서에 따른 파일 로딩
- 5) 만약 /home/test/public_html 디렉터리에 DirectoryIndex에 지시한 파일이 모두 없고 Options 옵션에 Indexes 값이 설정되어 있다면 디렉터리 파일리스트를 웹브라우저에 출력
- 6) 만약 Options 항목에 Indexes 값이 설정되어 있지 않다면 "Not Found" 라는 에러페이지 출력

- 가상호스트 설정

```
<VirtualHost *:80>  
    ServerAdmin webmaster@dummy-host.example.com  
    DocumentRoot /www/docs/dummy-host.example.com  
    ServerName dummy-host.example.com  
    ErrorLog logs/dummy-host.example.com-error_log  
    CustomLog logs/dummy-host.example.com-access_log common  
</VirtualHost>
```

가상호스트 설정부분은 초기에 주석처리되어있어 주석 해제 후 사용 CustomLog 옵션은 해당 가상호스트의 로그파일 경로 지정



5. 기능소개



5.2 httpd 다양한 옵션 및 기능(1/4)

- 유용한 httpd 명령어 옵션

-d serverroot

ServerRoot 지시문의 초기값을 serverroot 값으로 설정, 기본은 /etc/httpd

-f config

데몬 시작 시 config 파일의 지시문 사용, config가 / 로 시작하지 않으면(절대경로가 아니면) ServerRoot 와 관련된 경로로 간주, 기본값은 conf/httpd.conf

-k start|restart|graceful|stop|graceful-stop

httpd 데몬 시작, 다시 시작 또는 중지

-e level

서버 시작 시 LogLevel 설정, 이 기능은 시작시 문제를 찾기 위해 오류 메시지의 자세한 표시를 일시적으로 늘리는 데 유용

-S

설정된 가상호스트 내용 출력



5. 기능소개



5.2 httpd 다양한 옵션 및 기능(2/4)

-t

설정 파일의 문법 점검 (httpd.conf)

문법에 문제가 없을 경우 "Syntax OK" 구문 출력

-E file

지정된 파일에 로그 저장

```
[root@localhost ~]# /usr/sbin/httpd -E /root/httpd.log
[root@localhost ~]# ps -ef | grep httpd
root      1200      1    0 09:21 ?        00:00:00 /usr/sbin/httpd -E /root/httpd.log
apache    1201    1200    0 09:21 ?        00:00:00 /usr/sbin/httpd -E /root/httpd.log
apache    1202    1200    0 09:21 ?        00:00:00 /usr/sbin/httpd -E /root/httpd.log
apache    1203    1200    0 09:21 ?        00:00:00 /usr/sbin/httpd -E /root/httpd.log
apache    1204    1200    0 09:21 ?        00:00:00 /usr/sbin/httpd -E /root/httpd.log
apache    1205    1200    0 09:21 ?        00:00:00 /usr/sbin/httpd -E /root/httpd.log
root      1207    1094    0 09:21 pts/0    00:00:00 grep --color=auto httpd
[root@localhost ~]# ls /root/httpd.log
/root/httpd.log
```

[E 옵션으로 로그파일 지정]

* 위 그림에서 프로세스 확인하여 옵션이 적용된 것을 확인, 실제 에러로그 파일이 설정경로 대로 생성



5. 기능소개



5.2 httpd 다양한 옵션 및 기능(3/4)

- v
데몬 버전을 출력
- V
데몬의 버전 및 빌드 매개변수 출력 (컴파일 설치내용)
- l
웹서버에 설치되어 있는 아파치 모듈 확인
- L
웹서버 설정 지시자 확인

```
[root@localhost local]# httpd -v
Server version: Apache/2.4.6 (CentOS)
Server built:   Oct 11 2017 17:16:47
```

[httpd -v 옵션 사용]

```
[root@localhost local]# httpd -V
AH00558: httpd: Could not reliably determine the server's
his message
Server version: Apache/2.4.6 (CentOS)
Server built:   Oct 11 2017 17:16:47
Server's Module Magic Number: 20120211:24
Server loaded: APR 1.4.8, APR-UTIL 1.5.2
Compiled using: APR 1.4.8, APR-UTIL 1.5.2
Architecture:  64-bit
Server MPM:     prefork
                threaded:  no
                forked:   yes (variable process count)
Server compiled with....
-D APR_HAS_SENDFILE
-D APR_HAS_MMAP
-D APR_HAVE_IPV6 (IPv4-mapped addresses enabled)
-D APR_USE_SYSVSEM_SERIALIZE
-D APR_USE_PTHREAD_SERIALIZE
-D SINGLE_LISTEN_UNSERIALIZED_ACCEPT
-D APR_HAS_OTHER_CHILD
-D AP_HAVE_RELIABLE_PIPED_LOGS
-D DYNAMIC_MODULE_LIMIT=256
-D HTTPD_ROOT="/etc/httpd"
-D SUEXEC_BIN="/usr/sbin/suexec"
-D DEFAULT_PIDLOG="/run/httpd/httpd.pid"
-D DEFAULT_SCOREBOARD="logs/apache_runtime_status"
-D DEFAULT_ERRORLOG="logs/error_log"
-D AP_TYPES_CONFIG_FILE="conf/mime.types"
-D SERVER_CONFIG_FILE="conf/httpd.conf"
```

[httpd -V 옵션 사용]

```
[root@localhost local]# httpd -l
Compiled in modules:
  core.c
  mod_so.c
  http_core.c
```

[httpd -l 옵션 사용]

```
[root@localhost local]# httpd -L | grep VirtualHost
AH00558: httpd: Could not reliably determine the server's
localhost.localdomain. Set the 'ServerName' directive global
<VirtualHost (core.c)
NameVirtualHost (core.c)
```

[httpd -L 옵션 사용]



5. 기능소개



5.2 httpd 다양한 옵션 및 기능(4/4)

-h

데몬이 사용할 수 있는 옵션 도움말 출력 대부분의 데몬, 명령어에서 --help 나 -h 옵션으로 도움말을 볼 수 있으며, 실제로 명령어별로 데몬별로 많은 옵션들이 존재하여 유용하게 옵션 및 용도를 확인할 수 있다.

-h 옵션보다 자세한 설명은 man 명령어로 볼 수 있다.
man 명령어, man 데몬경로
예) man cp, man /usr/sbin/httpd

```
[root@localhost ~]# /usr/sbin/httpd -h
Usage: /usr/sbin/httpd [-D name] [-d directory] [-f file]
                    [-C "directive"] [-c "directive"]
                    [-k start|restart|graceful-stop|stop]
                    [-v] [-V] [-h] [-l] [-L] [-t] [-T] [-S] [-X]

Options:
-D name           : define a name for use in <IfDefine name> directives
-d directory     : specify an alternate initial ServerRoot
-f file          : specify an alternate ServerConfigFile
-C "directive"   : process directive before reading config files
-c "directive"   : process directive after reading config files
-e level         : show startup errors of level (see LogLevel)
-E file          : log startup errors to file
-v              : show version number
-V              : show compile settings
-h              : list available command line options (this page)
-l              : list compiled in modules
-L              : list available configuration directives
-t -D DUMP_VHOSTS : show parsed vhost settings
-t -D DUMP_RUN_CFG : show parsed run settings
-S              : a synonym for -t -D DUMP_VHOSTS -D DUMP_RUN_CFG
-t -D DUMP_MODULES : show all loaded modules
-M              : a synonym for -t -D DUMP_MODULES
-t              : run syntax check for config files
-T              : start without DocumentRoot(s) check
-X              : debug mode (only one worker, do not detach)
```

[httpd -h 옵션 사용]



5. 기능소개



5.3 가상호스트 기능(1/2)

- 가상호스트 (VirtualHost) 란?

가상호스트란 하나의 웹서버에서 여러 개의 웹서비스를 사용하게 할 수 있는 서비스 호스팅 업체의 경우 하나의 서버에 많게는 100개가 넘는 가상호스트를 설정하여 서비스하기도 한다.

- 아파치에서 적용할 수 있는 가상호스트 종류

이름 기반 가상호스트 : 같은 IP에서 도메인명이 다른 가상호스트 운영 (가장 보편적)

IP 기반 가상호스트 : 가상호스트별로 IP 각각 부여

포트 기반 가상호스트 : 동일한 호스트에 포트만 다르게 지정

* 기본 가상호스트 : 특정 호스트에 해당 사항이 없을 때 기본적으로 응답하는 호스트

- 이름 기반 가상호스트와 IP 기반 가상호스트 비교

이름 기반 – 하나의 IP 로 여러 개의 가상호스트 사용

클라이언트 요청 시 전송하는 헤더 중 Host 헤더정보를 사용하여 가상호스트 설정에 따라 서비스

IP 기반 – 서버에 호스트별 IP 주소 사용

IP를 추가하기 위해 네트워크 카드 추가 또는 별도 네트워크 설정이 필요



5. 기능소개



5.3 가상호스트 기능(2/2)

- 예제를 통한 비교

이름 기반

```
Listen 80 NameVirtualHost *:80
```

```
<VirtualHost *:80>
```

```
    ServerName site1.example.com  
    ServerAlias site1.co.kr ServerAlias site.kr  
    DocumentRoot /var/www/site1
```

```
</VirtualHost>
```

```
<VirtualHost *:80>
```

```
    ServerName site2.example.com  
    DocumentRoot /var/www/site2
```

```
</VirtualHost>
```

IP 기반

```
Listen 192.168.111.100:80  
Listen 192.168.111.200:80
```

```
<VirtualHost 192.168.111.100:80>
```

```
    DocumentRoot /var/www/site1  
    ServerName site1.example.com
```

```
</VirtualHost>
```

```
<VirtualHost 192.168.111.200:80>
```

```
    DocumentRoot /var/www/site2  
    ServerName site2.example.com
```

```
</VirtualHost>
```

TIP

실제 IP 주소를 아낄 수 있고, 네트워크 설정이나 네트워크 카드 추가가 필요 없는 이름 기반 가상호스트 설정을 많이 사용한다.



5. 기능소개



5.4 가상호스트 설정(1/5)

1. httpd.conf 파일 수정

메인 설정파일은 설치 위치에 따라 달라질 수 있음

[목차 2.3 설치 및 실행 (Linux compile)] 부분에서 아파치 최신버전으로 컴파일한 내용으로 진행

아파치 2.4.x 버전 이하 버전에서의 설정과 다를 수 있음

가상호스트 설정파일은 httpd-vhosts.conf 사용

```
# vi /usr/local/httpd-2.4.28/conf/httpd.conf
```

```
NameVirtualHost *:80 ← 추가 (Apache 2.4.x 이상에선 추가하지 않아도 됨)
```

```
#Include conf/extra/httpd-vhosts.conf 주석 제거 (# 기호 제거)
```

아래 주석 제거 후 Require all denied 를 Require all granted 로 수정

```
#<Directory /> #All
```

```
owOverride none #
```

```
Require all denied #
```

```
</Directory>
```



5. 기능소개



5.4 가상호스트 설정(2/5)

2. httpd-vhosts.conf 파일 수정

왼쪽 그림과 같이 수정

각 옵션에 대한 내용은 [목차 5.1 httpd.conf 설정 파일] 부분참고

3. 홈페이지 경로 및 테스트 페이지 생성

```
# mkdir /home/test
```

```
# mkdir /home/test2
```

```
# vi /home/test/index.html "This is webtest.com page" 추가 #
```

```
vi /home/test2/index.html
```

```
"This is webtest2.com page" 추가
```

4. 디렉터리 접근권한 부여

```
# chmod -R 755 test
```

```
# chmod -R 755 test2
```

```
#  
# VirtualHost example:  
# Almost any Apache directive may go into a VirtualHost container.  
# The first VirtualHost section is used for all requests that do not  
# match a ServerName or ServerAlias in any <VirtualHost> block.  
#  
<VirtualHost *:80>  
#   ServerAdmin webmaster@dummy-host.example.com  
   DocumentRoot "/home/test"  
   ServerName webtest.com  
   ServerAlias www.webtest.com  
#   ErrorLog "logs/dummy-host.example.com-error_log"  
#   CustomLog "logs/dummy-host.example.com-access_log" common  
</VirtualHost>  
  
<VirtualHost *:80>  
#   ServerAdmin webmaster@dummy-host2.example.com  
   DocumentRoot "/home/test2"  
   ServerName webtest2.com  
   ServerAlias www.webtest2.com  
#   ErrorLog "logs/dummy-host2.example.com-error_log"  
#   CustomLog "logs/dummy-host2.example.com-access_log" common  
</VirtualHost>
```

[httpd-vhosts.conf 파일 내용]



5. 기능소개



5.4 가상호스트 설정(3/5)

5. 아파치 재시작

```
# /usr/local/httpd-2.4.28/bin/httpd -k restart
```

```
# ps -ef | grep httpd
```

6. 접속할 클라이언트 호스트 파일 수정 (사용가능 도메인이 없을 경우)

서버에서 설정한 도메인이 사용가능 도메인이 아니므로 접속할 클라이언트의 호스트파일 수정

* 실제 구매, 사용하는 FQDN 도메인 입력 시 해당 부분 패스

윈도우 PC의 경우 권한문제로 hosts 파일을 수정하지 못하므로 아래와 같이 접근

[명령 프롬프트를 관리자 권한으로 실행 > "notepad C:\Windows\System32\drivers\etc\hosts" 입력 >

Hosts 파일에 아래내용 추가 후 저장

192.168.11.230	webtest.com
192.168.11.230	www.webtest.com
192.168.11.230	webtest2.com
192.168.11.230	www.webtest2.com



5. 기능소개



5.4 가상호스트 설정(4/5)

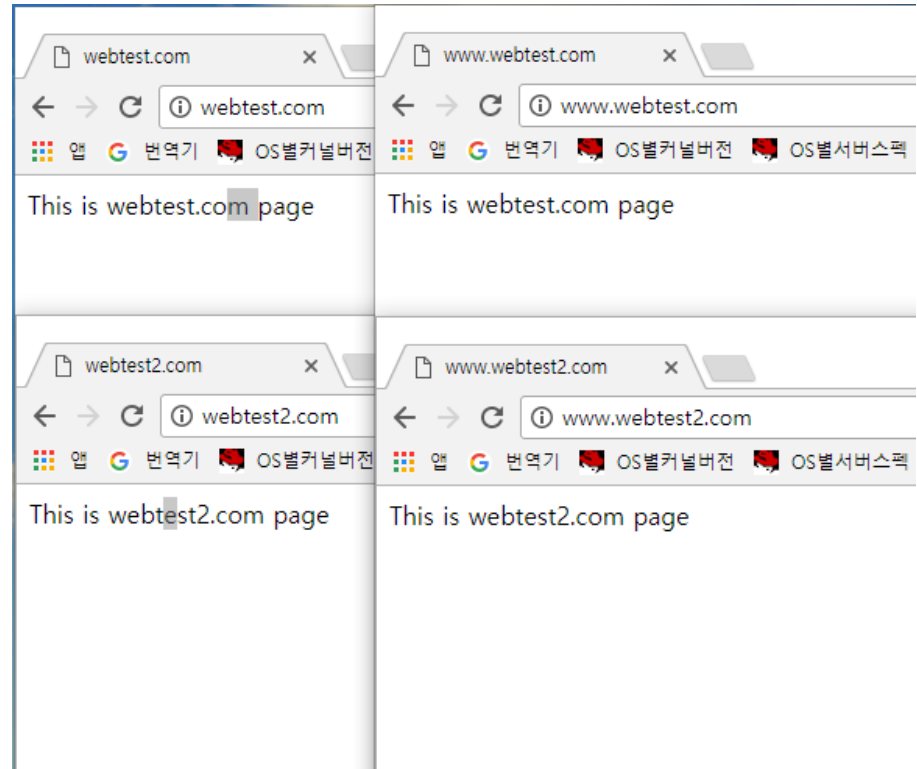
7. 웹브라우저에서 서버 도메인으로 접속, 웹페이지 확인

```
cmd 선택 관리자: 명령 프롬프트
C:\Windows\System32>notepad C:\Windows\System32\drivers/etc/hosts
C:\Windows\System32>

hosts - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
# Additionally, comments (such as these) may be inserted on
individual
# lines or following the machine name denoted by a '#'
symbol.
#
# For example:
#
#      102.54.94.97      rhino.acme.com      # source
server
#      38.25.63.10     x.acme.com          # X
client host

# localhost name resolution is handled within DNS itself.
#      127.0.0.1       localhost
#      ::1             localhost
192.168.11.230 webtest.com
192.168.11.230 www.webtest.com
192.168.11.230 webtest2.com
192.168.11.230 www.webtest2.com
```

[윈도우PC에서 hosts 파일 수정]



[웹브라우저에서 사이트 확인]



5. 기능소개



5.4 가상호스트 설정(5/5)

httpd 옵션 -S 로 VirtualHost 설정 확인

```
# /usr/local/httpd-2.4.28/bin/httpd -S
```

```
[root@localhost ~]# /usr/local/httpd-2.4.28/bin/httpd -S
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using localhos
VirtualHost configuration:
*:80
    is a NameVirtualHost
    default server webtest.com (/usr/local/httpd-2.4.28/conf/extra/httpd-vhosts.conf:23)
    port 80 namevhost webtest.com (/usr/local/httpd-2.4.28/conf/extra/httpd-vhosts.conf:23)
    alias www.webtest.com
    port 80 namevhost webtest2.com (/usr/local/httpd-2.4.28/conf/extra/httpd-vhosts.conf:32)
    alias www.webtest2.com
ServerRoot: "/usr/local/httpd-2.4.28"
Main DocumentRoot: "/usr/local/httpd-2.4.28/htdocs"
Main ErrorLog: "/usr/local/httpd-2.4.28/logs/error_log"
Mutex default: dir="/usr/local/httpd-2.4.28/logs/" mechanism=default
PidFile: "/usr/local/httpd-2.4.28/logs/httpd.pid"
Define: DUMP_VHOSTS
Define: DUMP_RUN_CFG
User: name="daemon" id=2
Group: name="daemon" id=2
```

[httpd -S 옵션 사용]



6. 활용예제

세부 목차



1. 웹인증 (.htaccess)
2. 웹인증 사이트 구축



6. 활용예제



6.1 웹인증 (.htaccess)

- **.htaccess 란?**

.htaccess 란 웹서버 소프트웨어 아파치(Apache)에서 정의하는 환경설정 파일
웹루트 디렉터리 하위의 디렉터리별로 서로 다른 환경설정을 구축하고자 할 때 .htaccess 설정을 달리하여
각 디렉터리에 추가한다.

- **웹인증 사용용도**

회원제로 운영하는 사이트, 특정그룹별로 접속 허용하는 사이트
일반인에겐 공개 페이지를 사내직원에게만 업무 페이지를 보여주는 사이트
운영자용 페이지 접속 시

- **서버 설정항목**

httpd.conf 파일 웹인증허용설정
웹인증할 디렉터리에서 .htaccess 파일 생성 사용자들
의 ID, Password 가 저장된 .htpasswd 파일 생성 웹브라
우저에서 웹인증테스트



6. 활용예제



6.2 웹인증 사이트 구축(1/4)

- 구축 환경
 - OS 버전 : CentOS 7.464bit
 - Apache 버전 : httpd-2.4.28
- 구축목표 : 특정 사용자만 인증하여 접속할 수 있는 사이트 구축
- 사전준비 : admin 이라는 사용자의 사이트 구축

useradd admin

grep admin /etc/passwd ← /etc/passwd 계정파일에서 admin 계정 생성 확인

```
[root@localhost ~]# useradd admin
[root@localhost ~]# grep admin /etc/passwd
admin:x:1001:1001::/home/admin:/bin/bash
```

[admin 계정 생성 및 확인]

vi /usr/local/httpd-2.4.28/conf/httpd.conf

수정 : DocumentRoot " /usr/local/httpd-2.4.28/htdocs" > DocumentRoot " /home/admin"

추가 : 수정한 DocumentRoot 열 아래 아래 내용 추가



6. 활용예제



6.2 웹인증 사이트 구축(2/4)

```
<Directory "/home/admin">
```

```
Options Indexes FollowSymLinks  
AllowOverride AuthConfig Require all granted
```

```
</Directory>
```

```
DocumentRoot "/home/admin"  
  
<Directory "/home/admin">  
    Options Indexes FollowSymLinks  
    AllowOverride AuthConfig  
    Require all granted  
</Directory>
```

Admin 계정 기본 페이지 생성

[httpd.conf 설정파일 admin 계정 설정]

```
# vi /home/admin/index.html
```

```
추가 : <font size=30><b>This is Admin Page</b></font>
```

/home/admin 디렉터리 이하 모든 파일/디렉터리 소유자, 소유그룹 admin 으로 변경

```
# chown admin.admin /home/admin/ -R
```

/home/admin 디렉터리에 권한 설정 (실행 권한 부여하여 디렉터리에 있는 문서를 읽기 위함)

```
# chmod 755 /home/admin/
```

```
# /usr/local/httpd-2.4.28/bin/httpd -k restart
```



6. 활용예제



6.2 웹인증 사이트 구축(3/4)

- 웹인증 사이트구축

httpd.conf 설정파일 모듈라인에 mod_auth_basic.so 확인

```
[root@localhost ~]# grep mod_auth_basic.so /usr/local/httpd-2.4.28/conf/httpd.conf
LoadModule auth_basic_module modules/mod_auth_basic.so
```

[httpd.conf 설정파일 mod_auth_basic.so 모듈 확인]

1.httpd.conf 설정파일 <Directory ~> 지시자에서 AllowOverride 옵션 값 AuthConfig 확인
사전 준비에서 설정

2. 웹인증할 디렉터리내에서의 .htaccess 파일생성

vi /home/admin/.htaccess

아래 내용 추가

// 이 글은 설명을 위한 글로 왼쪽 내용만 파일에 추가

AuthName "Admin-Auth" // AuthName 인증창의 이름, 보통 홈페이지 이름 입력

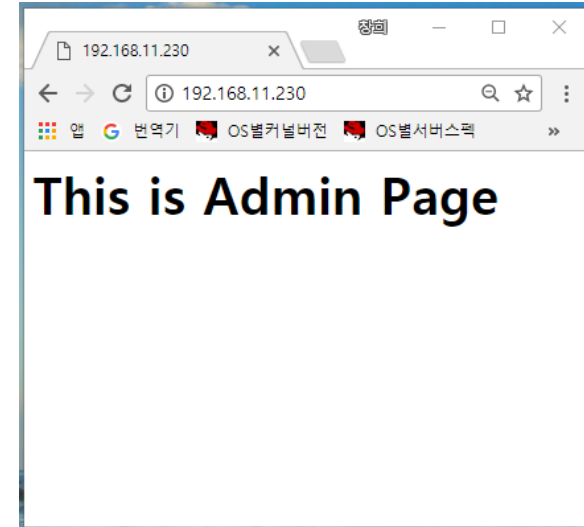
AuthType Basic // 인증타입 설정, 종류는 Basic 과 Digest가 있다.

AuthUserFile /home/admin/.htpasswd // 인증정보가 저장된 파일 위치

<Limit GET POST> // GET 방식과 POST 방식 접근만을 허용

Require valid-user // 패스워드로 인증된 사용자만이 접근 허용

</Limit>



[웹인증 구축 전 사이트 확인]



6. 활용예제



6.2 웹인증 사이트 구축(4/4)

3. httpd 패키지에 포함된 htpasswd 명령어를 통한 admin 계정의 인증정보 파일 생성

```
# /usr/local/httpd-2.4.28/bin/htpasswd -c /home/admin/.htpasswd admin
```

```
[root@localhost ~]# /usr/local/httpd-2.4.28/bin/htpasswd -c /home/admin/.htpasswd admin
New password:
Re-type new password:
Adding password for user admin
```

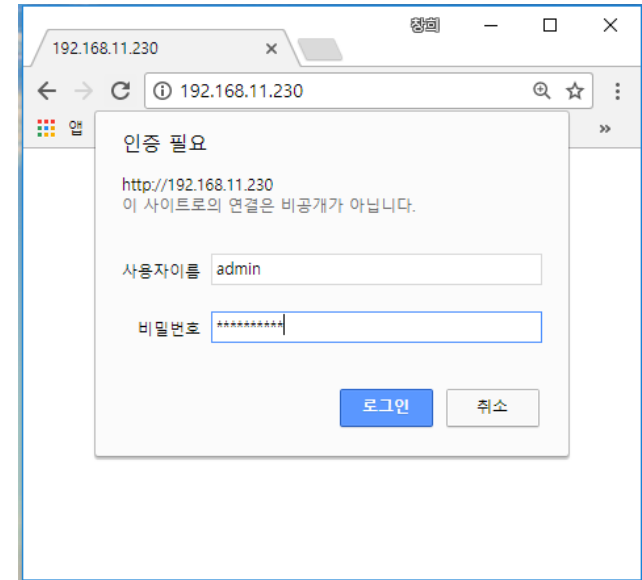
[인증정보파일 생성]

```
[root@localhost ~]# cat /home/admin/.htpasswd
admin:$apr1$P5BTFX0x$eDn8M.ORDNC5swt2rxFqq.
```

[인증파일 내용확인]

4. 암호인증 사이트 확인

사이트 접속 시 웹인증 창 확인, 인증정보 입력 시 정상접속 확인



[암호인증 사이트 확인]





Q APM을 많이 구축한다는데... APM 이 뭔가요?

&

A APM은 Apache + PHP + MySQL 의 줄임말 입니다. 이 3가지를 연동하여 운영하도록 만든 환경입니다. PHP는 웹프로그래밍 언어이며, HTML과 다르게 웹브라우저에서 소스를 해석하는 것이 아닌 서버에서 해석하여 웹브라우저에 전달합니다. MySQL 은 SQL 데이터베이스 서버입니다. 사진, 동영상, 게시판 등 훨씬 많은 데이터들을 저장/이용하기 위해 사용합니다.

Q Apache 사용률이 제일 높다고 하는데... 어떤 이유인가요?

&

A 많은 웹서버 소프트웨어가 있지만 Apache를 많이 사용하는 이유는 다음과 같습니다. 무료로 사용할 수 있으며, 오래 전부터 개발되어 안정적입니다. 이외에도 확장성, 튜닝을 통한 성능향상 등의 이유가 있으며 오랜 시간 많은 분들이 사용하여 다양한 예시와 참고자료가 있는 것도 이유 중 하나입니다.





Q Apache와 Tomcat 어떤 기능과 차이점이 있나요?

&

A Apache는 정적인 데이터를 처리하며, Tomcat는 동적인 데이터를 처리합니다. 물론 Tomcat은 정적인 데이터도 처리할 수 있지만 Apache보다 속도가 느립니다. 때문에 효율과 목적에 있어 동적인 데이터 처리에 사용합니다 보통 Apache와 Tomcat을 연동하여 사용합니다.

Q 정적인 데이터, 동적인 데이터가 어떻게 다르나요?

&

A 쉽게 말씀드리면 정적인 데이터는 결과값이 고정되어 있는 데이터, 동적인 데이터는 요청/상황 등 조건에 의해 결과값이 달라지는 데이터입니다. 예로 정적인 데이터는 이미지, 단순 HTML Page, 동적인 데이터는 날짜, 시간, 사용자의 연령에 따라 결과값이 다른 Page에서 볼 수 있습니다.



8. 용어정리



용어	설명
SSL	전송 계층 보안 TLS의 과거 명칭으로 현재도 많이 사용한다. 클라이언트/서버 응용 프로그램이 네트워크로 통신을 하는 과정에서 도청, 간섭, 위조를 방지하기 위해 설계되었다. 암호화로 인증, 통신 기밀성을 제공한다.
JSP	"Java Server Page" 약자. Java를 사용한 서버측 코드가 있는 웹페이지 확장자는 .jsp 를 사용한다.
Server Side	네트워크의 한 방식인 클라이언트-서버 구조에서 서버 쪽에서 행해지는 처리, 데이터 위조의 가능성을 줄일 수 있으며 민감한 데이터의 경우 (인터넷 뱅킹, 금융, 게임아이템) Server-Side로 처리해야 한다. 예로 JSP를 통한 Java가 있다.
Web Container	Servlet, JSP 파일, Server Side 언어코드가 포함된 파일들에 대한 요청 처리를 수행한다.
WAS	"Web Application Server" 약자. 웹서버와 웹컨테이너의 결합으로 단순 웹서버와 구별된다. 클라이언트의 요청이 있을때 내부 프로세스로 작업 후 이것을 다시 클라이언트에게 돌려주는 역할을 수행 한국에서는 일반적으로 "WAS" 영어권에서는 "AS" 로 불린다. 예로 Tomcat이 있다.



Open Source Software Installation & Application Guide



이 저작물은 크리에이티브 커먼즈 [저작자표시-비영리-동일조건 변경허락 2.0 대한민국 라이선스]에 따라 이용하실 수 있습니다.